



**DISEÑO DE UN SISTEMA DE ACCESO POR RECONOCIMIENTO DE HUELLA  
DACTILAR MEDIANTE LA PLATAFORMA ARDUINO**

**TRABAJO DE INVESTIGACIÓN FORMATIVA**

**JEISSON ELÍAS SÁCHICA PARRA**

**DIRECTOR**

**PhD(c). JIMMY ALEXANDER CORTÉS OSORIO**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA**

**FACULTAD DE TECNOLOGÍAS**

**CICLO II TECNOLOGÍA EN MECATRÓNICA**

**PEREIRA, RISARALDA**

**01 DE FEBRERO, 2018**

## INTRODUCCIÓN

Los sistemas de acceso restringido han sido parte de nuestras vidas desde inicios de nuestra civilización, y la necesidad de permitir el acceso a lugares o información solo a usuarios específicos ha estado siempre presente. Los métodos mecánicos como cuerdas, alambres o más recientemente, llaves metálicas, han sido de gran importancia en la historia de toda la humanidad, sin embargo, con el surgimiento de la era digital, nuevos métodos han nacido para realizar la misma tarea de una manera más rápida y sin requerir de objetos físicos, siendo uno de los más importantes, el reconocimiento de huella dactilar.

Desde hace siglos las características biométricas han sido objeto de estudio por la biología debido a que permiten el archivamiento y clasificación de especies de animales y plantas y determinar tendencias de evolución o adaptación que puedan presentar [1]. Los primeros usos de manera consciente de dichas características para la identificación y manejo de datos acerca de individuos se dieron a finales del siglo XIX y principios del siglo XX, cuando las investigaciones policiales empezaron a guardar registros de diferentes medidas fisiológicas tomadas de los criminales, entre las cuales se almacenaba una impresión de las huellas dactilares, y en 1906 se dio el primer caso en Estados Unidos en donde se sentenció a un ladrón a 7 años de prisión dada la identificación de su huella dactilar que fue tomada tiempo antes en Londres donde había cometido otro robo [2].

Desde entonces, los avances que se han hecho en el campo abarcan cada vez mayores complejidades y hacen que todo el proceso sea lo más simple para la interacción entre el usuario y el sistema de reconocimiento. Existen ya muchas variedades de sensores de huella dactilar por precios muy económicos [3] e incluso hay en el mercado pequeñas unidades que no solo son capaces de realizar la adquisición de la imagen biométrica sino también el preprocesamiento, caracterización y emparejamiento necesarios para poder guardar usuarios nuevos de acuerdo con una huella en específico, modificar usuarios ya existentes y por supuesto, reconocer cuando un usuario previamente almacenado es reconocido [4].

A pesar de que ya existen en el mercado diversos sistemas de acceso por reconocimiento de huella dactilar, el grado de intervención que permiten es muy bajo, y los requerimientos que se desean para la implementación empiezan a exigir demasiado para realizar la adaptación. Por esto surge la necesidad de realizar el diseño de un sistema compuesto con diversos módulos que cumplan cada función por separado e integrarlos mediante una plataforma programable como un dispositivo Arduino y así asegurar que se cumplan varias tareas en la entrada al laboratorio de visión por computadora, en donde por medio de un sensor de huella

dactilar, una pantalla de visualización, una interfaz de interacción y un módulo de conexión Wifi para el almacenamiento de información en un servidor web, se administre el acceso de personal autorizado e invitados.

En el presente documento se evidenciará detalladamente el proceso llevado a cabo por los integrantes y directores del grupo de investigación del laboratorio de aprendizaje automático y visión por computadora para completar el diseño de dicho sistema, utilizando un microcontrolador para realizar el procesamiento, un módulo Wifi para la comunicación con una base de datos en línea que almacena el registro de usuarios nuevos y el acceso de invitados al laboratorio, un sensor de huella dactilar que se encarga de reconocer si un usuario está o no registrado, una pantalla táctil que permite visualizar y recibir entradas de texto por parte de un usuario externo, dos módulos de comunicación por radiofrecuencia para juntar las secciones de control y potencia ubicadas al exterior y al interior del laboratorio y finalmente una etapa de potencia encargada de abrir o cerrar la cerradura de la puerta mediante un motor solenoide.

## **OBJETIVOS**

### **Objetivo general:**

Diseñar un sistema de acceso por reconocimiento de huella que no solo permita el ingreso a un laboratorio mediante el reconocimiento de usuarios existentes y una etapa de potencia, sino también el aprendizaje de usuarios nuevos y almacenamiento de registros de ingreso en línea.

### **Objetivos específicos:**

- Definir los parámetros del sistema general y los componentes a utilizar en cada etapa teniendo en cuenta las funcionalidades de cada dispositivo por separado para la evaluación de su aplicabilidad en el proyecto.
- Realizar los ajustes necesarios para verificar que los componentes sean los adecuados y cumplan con los requerimientos.
- Diseñar el esquemático y distribución física de los componentes del dispositivo integrado final de manera robusta, compacta y fácilmente accesible para reparación, reemplazo o actualización.

## CAPÍTULO 1. PLANTEAMIENTO DEL PROYECTO

El laboratorio de aprendizaje automático y visión por computadora de la universidad tecnológica de Pereira está ubicado en el centro de desarrollo vecinal (CDV) del barrio San Luis de la ciudad de Pereira. En este laboratorio se reúne el grupo de investigación que lleva el mismo nombre y que cuenta con una trayectoria de proyectos no solo apoyados sino realizados propiamente por integrantes y con el apoyo del grupo.

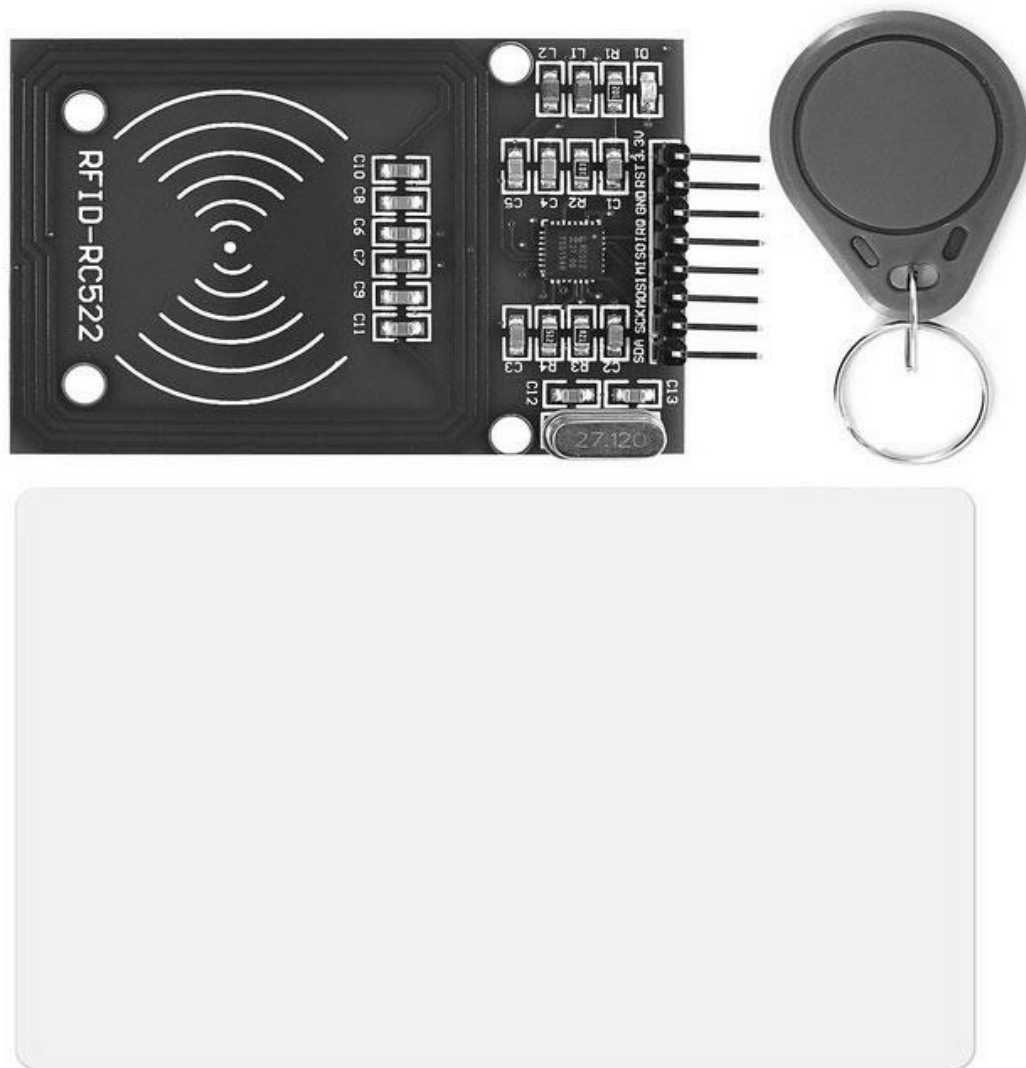
Anteriormente el laboratorio contó con un sistema de acceso diseñado e implementado por los estudiantes y directores basado en una pantalla LCD de 16x2, un teclado matricial para el ingreso de una contraseña y un microcontrolador 16F887 de Microchip permitiendo almacenar la contraseña actual en una memoria no volátil con el fin de que no se perdiera la contraseña configurada en caso de interrumpirse la alimentación de energía del microcontrolador y una etapa de potencia que abría o cerraba la cerradura de la puerta.



*Fig. 1 Ejemplo de un sistema integrado de acceso con clave. Fuente: [5]*

La ventaja de este sistema es que es relativamente simple de diseñar y construir pues su programación no es compleja y para este tipo de aplicaciones ya hay códigos estandarizados que cumplen con todas las funcionalidades requeridas para un sistema de acceso de este tipo.

Recientemente en el laboratorio de robótica aplicada ubicado en el mismo edificio, CDV, fue implementado por los integrantes del grupo un sistema de acceso basado en identificación por RFID haciendo uso de los carnés con los que cuentan cada estudiante, profesor, director o exalumno de la universidad, así se aprovecha el hecho de que cada persona ya cuenta con uno y no es necesario invertir capital en los mismos.



*Fig. 2 Sensor RFID y Tarjeta con chip integrado para lectura. Fuente: [6]*

Como primera instancia del proyecto, en una reunión de grupo se planteó la idea de implementar un sistema de acceso en el laboratorio más avanzado a comparación con el anterior que era con teclado e ingreso de contraseña, pero esta vez basándose en las opciones previamente documentadas, la idea era que el acceso no se basara en lo que se sabe (contraseña) o lo que se tiene (carné), sino en lo que se es (biometría) y por esto se decidió que un sistema de reconocimiento biométrico era la base y que por facilidad de programación, instrumentación disponible y recursos económicos, el reconocimiento de huella dactilar era la mejor opción. De esta manera, se dio base a las pautas para el proyecto:

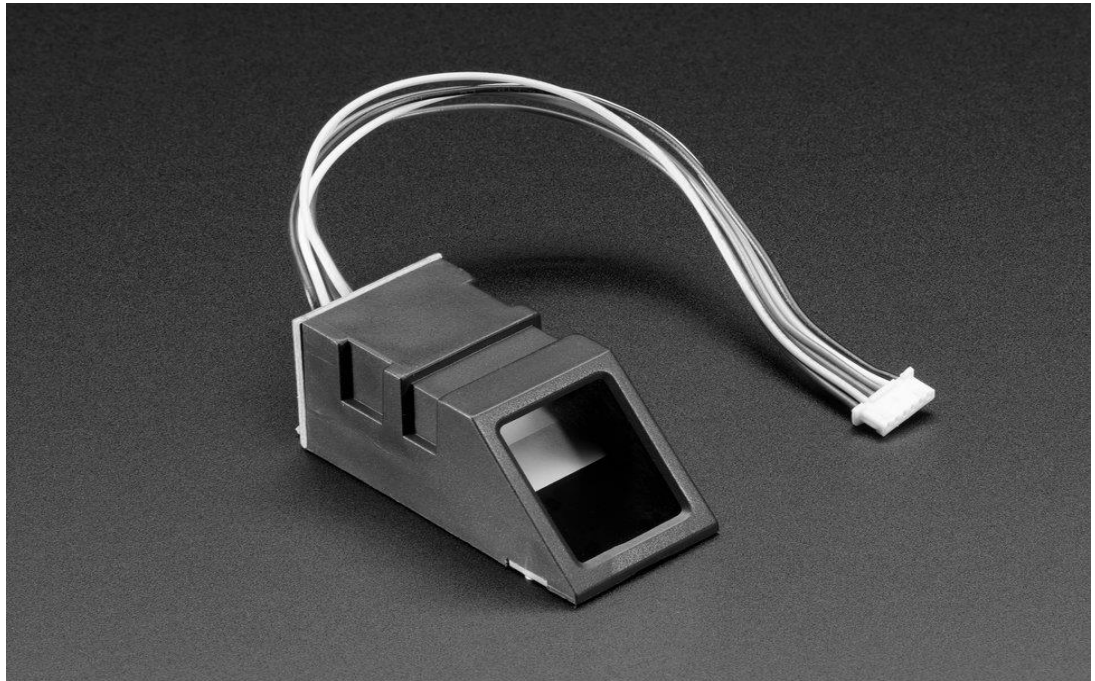
- Deberá contar con un sensor de huella dactilar para el reconocimiento de usuarios.
- Contará con un módulo Wifi que se encargue de realizar la conexión con una base de datos en línea y almacenar los registros de entrada o creación de usuarios nuevos.
- Se utilizará una pantalla táctil que permita la visualización del estado de un proceso de ingreso así mismo como la selección de opciones o entrada de texto.
- Para el procesamiento se utilizará un microcontrolador Arduino y su entorno de programación (IDE).
- Comunicación inalámbrica para aislar físicamente las secciones de control que irán ubicadas en la parte de afuera del laboratorio y la sección de potencia ubicada al interior de éste.
- Una etapa de potencia que al recibir la orden del circuito de control active un motor solenoide que abra o cierre la cerradura de la puerta.

Después de establecidos los anteriores puntos el siguiente paso consistió en seleccionar los dispositivos y componentes a utilizar, para esto se tuvo en cuenta la disponibilidad de estos en el laboratorio y facilidad de encontrarlos en el mercado de la ciudad con el fin de no tener que incurrir en gastos de envío y tiempos de espera en caso de tener que pedir desde otra ciudad o país.

La lista de componentes que se escogió finalmente fue la siguiente:

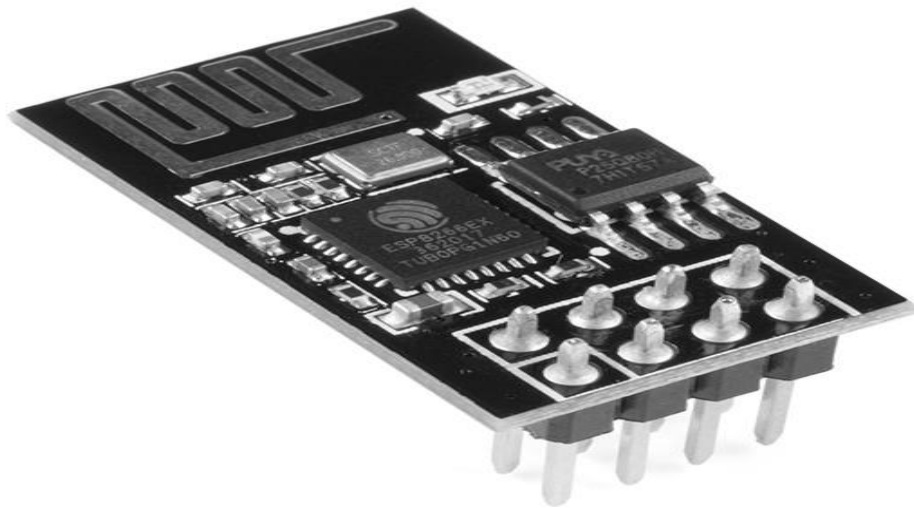
- ✓ El sensor escogido para realizar la lectura de la huella dactilar es un módulo lector 071405 que utiliza protocolo serial UART para poder comunicarse con cualquier microcontrolador fácilmente, este sensor no solo es capaz de tomar la imagen y generar la plantilla de la huella dactilar sino también de realizar el proceso de comparación y reconocimiento de la huella como perteneciente a un usuario previamente registrado y autorizado.





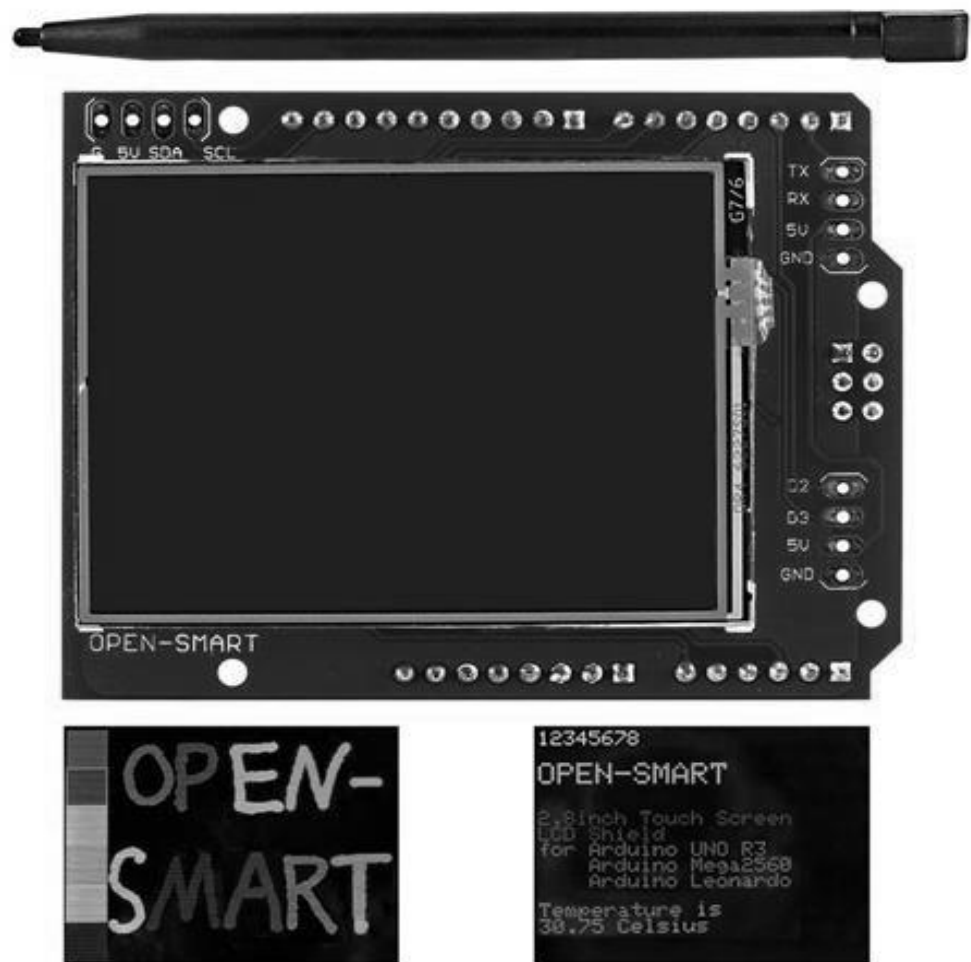
*Fig. 3 Sensor de huella dactilar con protocolo UART. Fuente: [4]*

- ✓ Para la comunicación entre el microcontrolador y la base de datos en línea existen dos opciones asequibles, la cuales son Ethernet y Wifi. Sin embargo, debido a la idea principal de que el sistema sea lo más compacto posible y aislado de cables o conexiones exteriores en lo posible se optó por hacer vía Wifi mediante el módulo ESP8266, muy simple y fácil de conseguir pero que a su vez permite hacer todas las acciones deseadas.



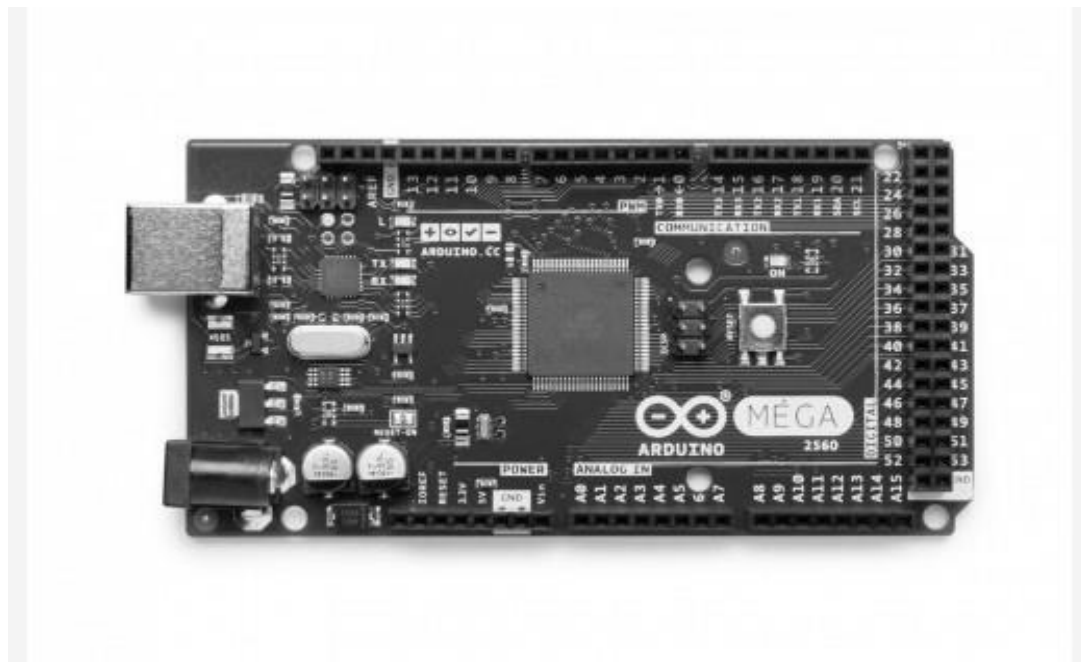
*Fig. 4 Módulo Wifi ESP8266-01 con antena. Fuente: [7]*

- ✓ Para la interfaz entre usuario y sistema se optó por una pantalla táctil LCD de 2,2 pulgadas de bajo costo pero que cuenta con buena calidad y tamaño apropiado para la aplicación, la razón por la que se escogió sobre la anterior pantalla LCD de 16x2 y el teclado matricial es porque el precio de la pantalla táctil es igual al precio combinado de la LCD tradicional y el teclado, otorga mucha más facilidad de manejo ya que utiliza menos pines que estos dos elementos y además brinda al sistema de un retoque moderno y robusto, haciéndolo ver como un producto comerciable y llamativo.



*Fig. 5 Pantalla LCD táctil de 2,2 pulgadas. Fuente: [8]*

- ✓ El microcontrolador escogido fue el Arduino Mega2560, se escogió este modelo sobre otros (Uno, Nano, Leonardo, etc.) debido a que los tres componentes anteriormente mencionados utilizan protocolo serial asíncrono UART para su respectiva comunicación y este modelo cuenta con 4 puestos físicos de comunicación serial disponibles para su uso paralelo y a altas velocidades.



*Fig. 6 Microcontrolador embebido Arduino Mega2560. Fuente: [9]*

De esta manera, teniendo definidos los elementos a utilizar se procedió con la siguiente fase del proyecto y adquirir toda la información importante o necesaria acerca de cada uno de ellos.

## CAPÍTULO 2. FUNCIONALIDAD DE LOS COMPONENTES

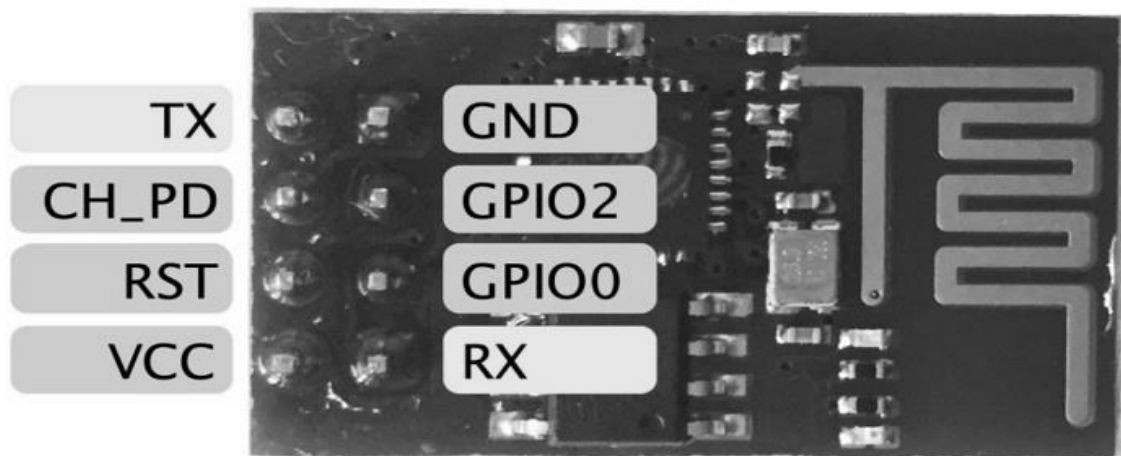
Para el desarrollo de este capítulo se contó con una serie de reuniones entre los integrantes del grupo a modo de dividir las tareas de investigación de cada componente y determinar el modo de funcionamiento, características y conexión de estos.

### Módulo Wifi

Este módulo es un integrado simple de 8 pines muy compacto y de pequeño tamaño que es capaz de funcionar tanto como un servidor como un punto de acceso. En su firmware de fábrica viene programado para ser configurado a través de comandos AT tal como si fuera un módem. A través de conexión serial con un microcontrolador o directamente desde un terminal serial, es posible enviar estos comandos AT y configurarlo fácilmente de acuerdo con las instrucciones del fabricante. Las especificaciones de este componente son las siguientes:

- ♦ Alimentación de 3.3V
- ♦ Protocolo Serial/UART (baudaje de hasta 115200 bps).
- ♦ Consumo de corriente regular de 70mA.
- ♦ Consumo de corriente máximo de 300mA.
- ♦ Protocolo TCP/IP integrado.
- ♦ Wifi b/g/n IEEE 802.11
- ♦ Modos de seguridad Wifi WPA y WPA2.
- ♦ Memoria flash de 1MB (8Mbit).
- ♦ Procesador de 32-bits (consumo de potencia en modo espera de <1mW).

La distribución de sus pines se expone en la siguiente imagen:



*Fig. 7 Distribución de pines en el módulo ESP8266-01. Fuente: [10]*

Como se observa en la imagen anterior, el módulo cuenta con:

- Dos pines de alimentación 3V3, los cuales son *GND* y *VCC*.
- Dos pines para la comunicación serial asíncrona UART, *TX* (*transmisión*) y *RX* (*recepción*).
- Dos pines de entrada o salida digital para propósito general, *GPIO0* y *GPIO2* (*general purpose in/out*).
- Un pin de habilitación de la antena, *CH\_PD*, que debe estar conectado a alimentación para poder activarla y tener un funcionamiento normal.
- Un pin de reset, *RST*, que reinicia el módulo por completo al presentarse un flanco de subida.

Para poner en funcionamiento el módulo ESP01 se requiere conectar el módulo con el microcontrolador a través de los pines *TX* y *RX*. Debido a que el módulo trabaja a un voltaje de 3V3 es necesario que el *TX* del microcontrolador que va al *RX* del módulo se conecte a un divisor de voltaje con tres resistencias del mismo valor, de esta manera el voltaje 5V proveniente del microcontrolador será reducido al voltaje 3V3 que soporta el módulo. Debido a que el voltaje 3V3 del módulo se encuentra dentro del rango TTL que maneja el microcontrolador, no es necesario amplificar este voltaje y se puede conectar directamente el *TX* del módulo al *RX* del microcontrolador. Además de esto las tierras de ambos dispositivos deben estar conectadas, y el pin *CH\_PD* del módulo conectado a *VCC* para habilitar la antena de este. Teniendo esto en cuenta, la conexión para la puesta en funcionamiento del módulo es la siguiente:

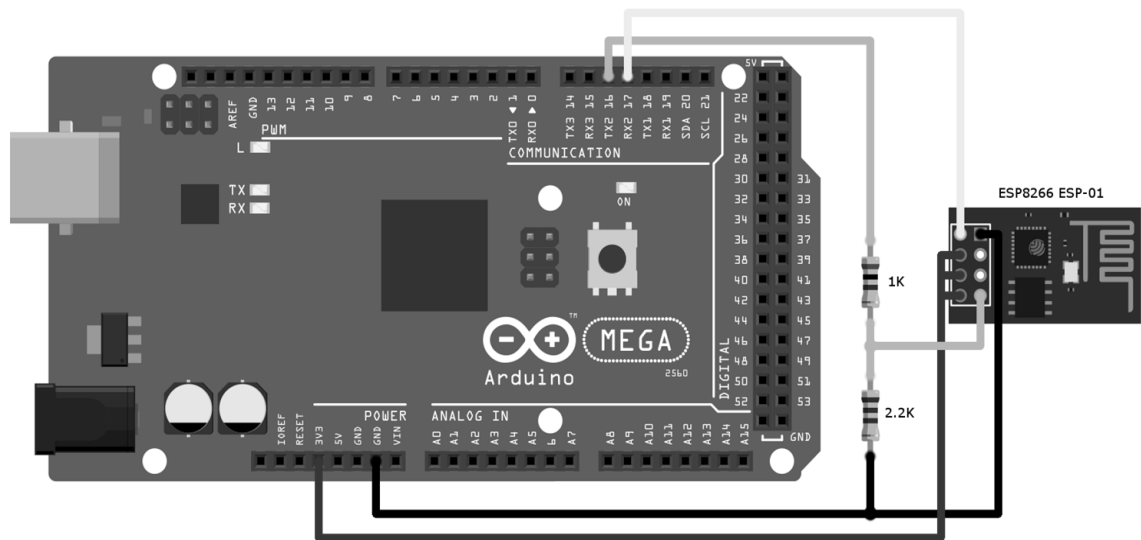


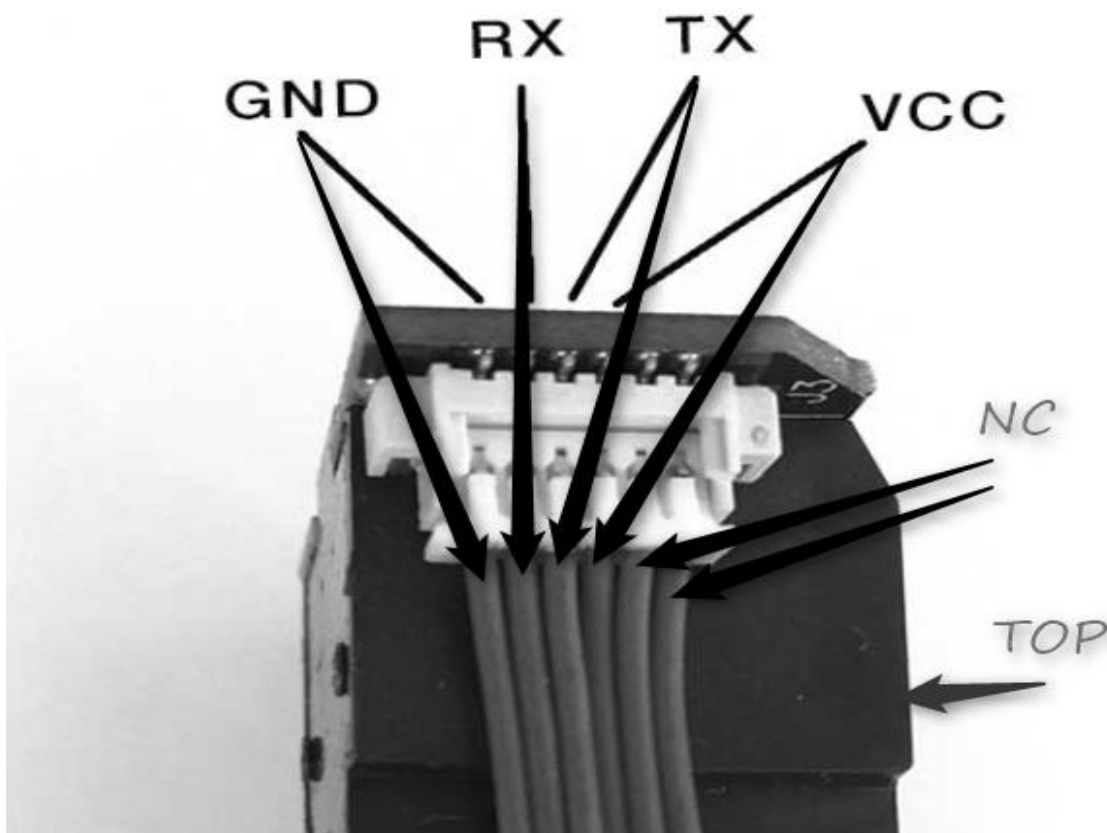
Fig. 8 Conexión del módulo ESP8266-01 con el Arduino. Fuente: [11]

### Sensor de huella dactilar

Este escáner es un componente que consta de un lente óptico y una unidad DSP que se encarga de hacer la renderización de la imagen, cálculo, hallazgo de características y emparejamiento de una plantilla con una existente en la memoria flash, la cual permite guardar hasta 162 huellas. Las especificaciones de este sensor son:

- ♦ Alimentación de 3.6–6.0 V.
- ♦ Consumo de corriente regular de 120mA.
- ♦ Consumo de corriente máximo de 150mA.
- ♦ Toma de imagen de <1s.
- ♦ Protocolo serial/UART (baudaje de hasta 57600bps).
- ♦ Tasa de falsos positivos de <0.001%.
- ♦ Tasa de falso negativo de <1%.
- ♦ Tamaño de archivo de firma de 256 bytes.
- ♦ Tamaño de archivo de plantilla de 512 bytes.

La distribución de los pines y cables que componen al sensor es la siguiente:



*Fig. 9 Distribución de los pines en el sensor óptico. Fuente: [4]*

Así podemos apreciar que este sensor cuenta en cuanto a hardware con 4 pines de los cuales:

- Dos pines de alimentación 5V, los cuales son *GND* y *VCC*.
- Dos pines para la comunicación serial asíncrona UART, *TX* (*transmisión*) y *RX* (*recepción*).
- Dos pines restantes que pertenecen a señales del sensor de proximidad integrado en caso de requerir una aplicación con esta.

Para el funcionamiento del sensor se cuenta con una librería completa de Arduino que la provee el fabricante y para la cual solo se requiere su descarga e instalación y realizar la conexión apropiada. Así, debido a que los rangos de operación del módulo lo permiten y para mayor facilidad en el manejo de fuentes, se alimentará el sensor con los mismos 3.3V del módulo Wifi y por lo tanto la conexión entre los terminales de transmisión y recepción del microcontrolador y el sensor es igual a la utilizada con el módulo Wifi, además, finalmente se conecta el pin de alimentación a 3.3V y se conectan las tierras en ambos dispositivos. Obtenemos entonces la conexión entre el microcontrolador y el sensor así:

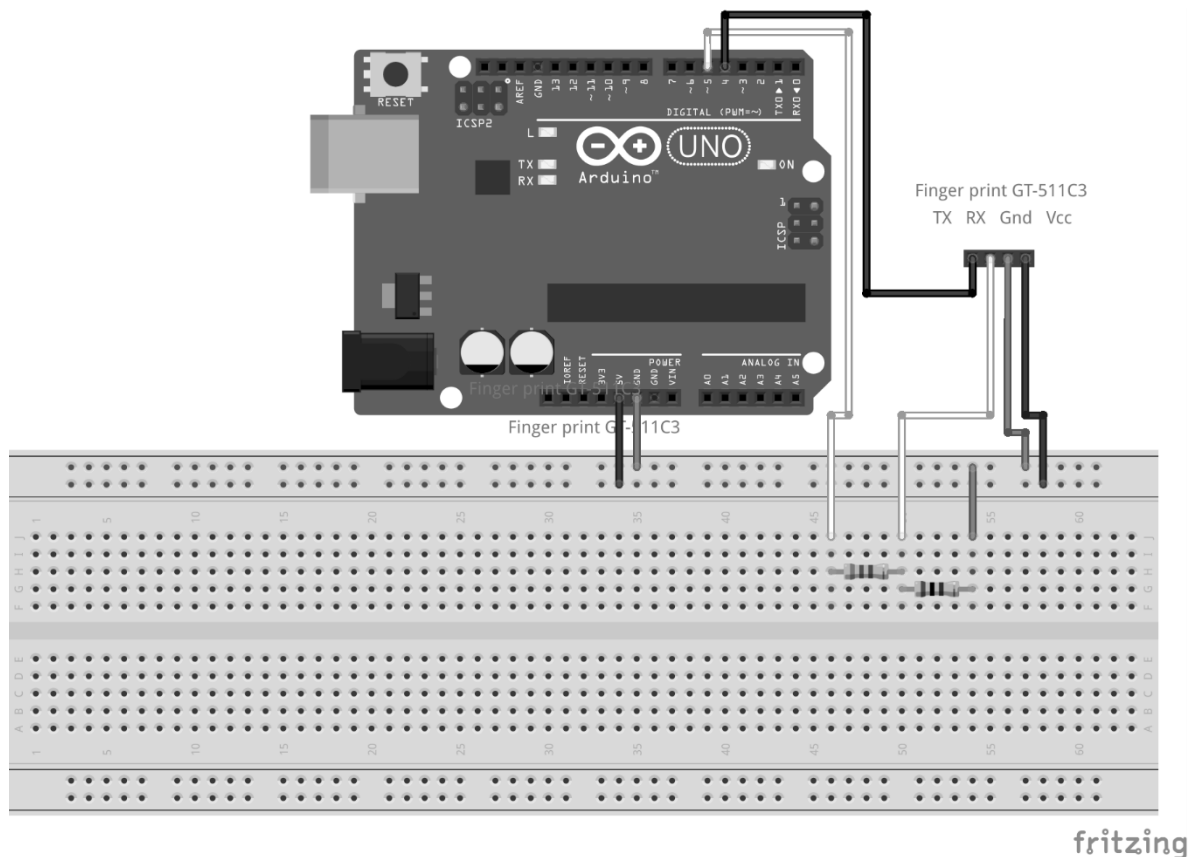


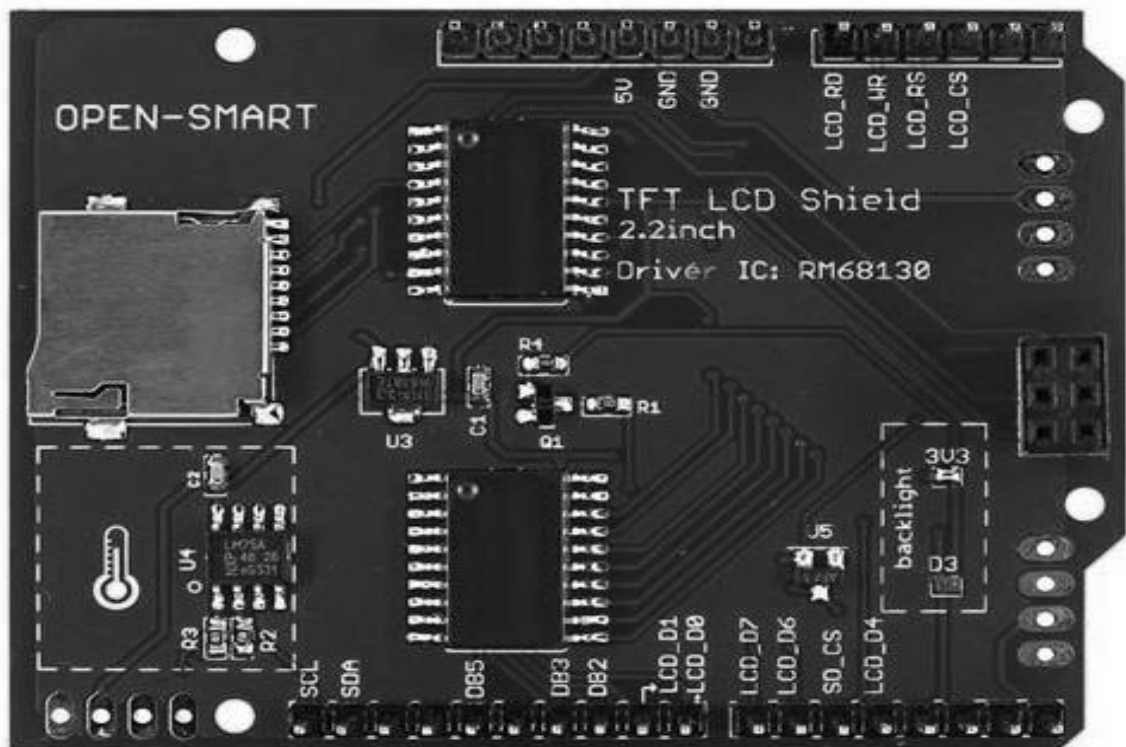
Fig. 10 Conexión de funcionamiento entre Arduino y el sensor. Fuente: [12]

## Pantalla LCD táctil

La pantalla que se escogió para el proyecto es ideal en cuanto a la relación entre el costo y la funcionalidad, cuenta con un tamaño adecuado para la visualización de un menú de ingreso, entrada de texto y reproducción de imágenes de usuarios registrados, además de que permite la escritura por medio de un teclado mediante el uso de su superficie táctil. Las especificaciones de este dispositivo son expresadas a continuación:

- ♦ Alimentación de 5V.
- ♦ Pantalla de 2.2 pulgadas.
- ♦ Consumo de corriente máximo 150mA.
- ♦ Resolución de 176x220 pixeles.
- ♦ Pantalla táctil resistiva.
- ♦ Conexión integrada con tarjeta de memoria microSD.
- ♦ Escala de color de 18-bits (262.000 colores distintos).
- ♦ Compatibilidad con Arduino Mega, Uno y Leonardo.

La distribución de los pines con los cuales cuenta la pantalla en hardware es de la siguiente manera:



*Fig. 11 Distribución de pines en la pantalla táctil LCD. Fuente: [8]*



En la imagen anterior se observa que la pantalla cuenta con una cantidad numerosa de pines, de los cuales los que serán utilizados se explican a continuación:

- Tres pines relacionados con la alimentación de 5V de la pantalla, los cuales son *GND* y *VCC*.
- Dos pines para comunicación I2C, *SDA* y *SCL*, que se puede efectuar con cualquier dispositivo que soporte este protocolo como sensores, módulos de reloj en tiempo real u otros.
- Dos pines para comunicación serial asíncrona UART, *TX* y *RX*.
- Pines para comunicación SPI requerida para la lectura y/o escritura de la tarjeta microSD.

Para colocar en funcionamiento la pantalla y darle el uso que se requiere, la conexión de esta es muy sencilla puesto que ya viene diseñada como una especie de *shield*, permitiendo su posicionamiento directo sobre la placa del Arduino Mega2560. La conexión entre la pantalla y el microcontrolador es expresada en la siguiente figura:



*Fig. 12 Conexión de la pantalla táctil LCD y el Arduino Mega2560. Fuente: [8]*

### CAPÍTULO 3. EVALUACIÓN DE LA APLICABILIDAD DE LAS PRINCIPALES PARTES

En este capítulo se desarrolló toda la parte de código, cableado, esquema y diseño de todos los componentes principales, así como de las dos etapas faltantes y que fueron finalizadas acá, estas son la etapa de comunicación inalámbrica entre exterior e interior y la etapa de potencia en el interior.

Para la finalización de la sección del módulo Wifi, se estableció una secuencia de comandos que serían enviados a este para asegurar que en cada reinicio del sistema general se garantice la conexión con la red y la base de datos, con este fin se utilizaron la siguiente serie de comandos AT en el módulo ESP8266-01 de acuerdo con el manual de instrucciones del fabricante \*12:

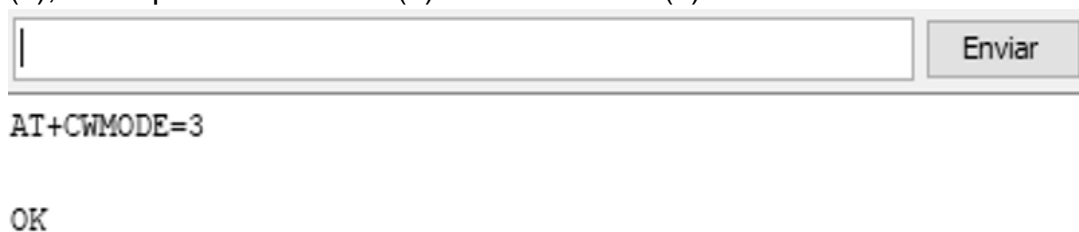
- Comando *AT*: Este comando se utiliza para realizar un test rápido de conexión entre el microcontrolador y el módulo, si la respuesta es un *OK* significa que hay comunicación bidireccional entre los dos dispositivos de manera correcta.



A screenshot of a serial terminal window. At the top, there is a text input field and a button labeled 'Enviar'. Below the input field, the command 'AT' is shown in blue text, and the response 'OK' is shown in green text.

*Fig. 13 Instrucción AT enviada al módulo y respuesta. Fuente: autor*

- Comando *AT+CWMODE=<mode>*: Este comando es el encargado de configurar el modo de trabajo del módulo, el cual puede ser como estación (1), como punto de acceso (2) o como ambas (3).



A screenshot of a serial terminal window. At the top, there is a text input field and a button labeled 'Enviar'. Below the input field, the command 'AT+CWMODE=3' is shown in blue text, and the response 'OK' is shown in green text.

*Fig. 14 Comando de configuración del modo de trabajo. Fuente: autor*

- Comando *AT+CWJAP=<ssid>,<password>*: Este comando se utiliza para establecer la comunicación entre el módulo Wifi y la red de internet inalámbrica local especificada.

AT+CWJAP="PISO\_3", "██████████"

OK

Fig. 15 Comando de conexión a la red inalámbrica local. Fuente: autor

Una vez se ha establecido la conexión entre el microcontrolador y el módulo y entre el módulo y la red local, existen dos funciones creadas para el ingreso de datos en la base de datos, la primera función se encarga de ingresar un registro de entrada cualquiera con el ID del usuario provisto por el sensor de huella dactilar al microcontrolador de la siguiente manera:

- Comando `AT+CIPSTART=<type>,<remote IP>,<remote port>`: Este comando se utiliza para iniciar la conexión entre el módulo Wifi y el servidor especificado una vez se está conectado a la red local.

AT+CIPSTART="TCP", "192.168.0.103", 80

CONNECT

OK

CLOSED

Fig. 16 Comando para inicio de conexión con el servidor. Fuente: autor

- Comando `AT+CIPSEND=<length>`: Este comando se encarga de comunicar al servidor el tamaño de la trama de datos (en bytes) que será enviada.

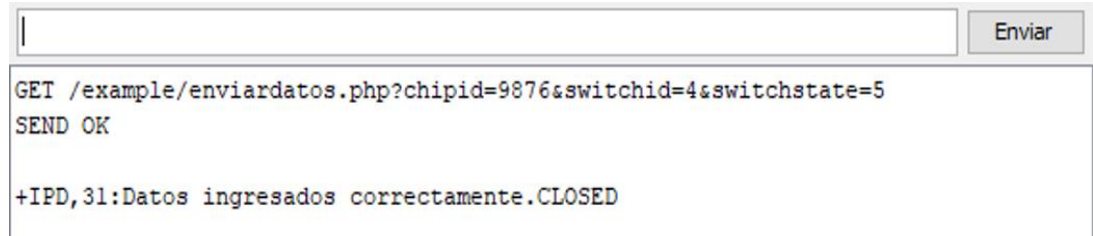
AT+CIPSEND=67

OK

>

Fig. 17 Comando de especificación del tamaño de la trama. Fuente: autor

- *>Trama de datos:* Una vez el módulo retorna el símbolo “>” lo siguiente y último que se envía es la trama de datos. De acuerdo con el servidor que se ha instalado localmente en la computadora la trama de datos se compone de una instrucción *GET* seguida de la dirección en la que está alojado el archivo *.php* que se encarga de introducir los datos al servidor y finalmente contiene las variables que el código del archivo requiere.



The screenshot shows a web browser window. At the top, there is an address bar with a cursor. To the right of the address bar is a button labeled "Enviar". Below the address bar, the browser's status bar displays the following text: "GET /example/enviardatos.php?chipid=9876&switchid=4&switchstate=5", "SEND OK", and "+IPD,31:Datos ingresados correctamente.CLOSED".

*Fig. 18 Trama de datos para ingresar un registro de entrada. Fuente: autor*

Para el sensor de huella dactilar, su operación se realizó mediante la librería provista por el equipo de desarrollo de *Adafruit*, esta permite de manera fácil e intuitiva realizar las operaciones básicas necesarias para controlar el sensor y obtener resultados calculados por éste. Esta librería consta de 3 principales funciones descritas a continuación:

- Función *getFingerprintIDez*: Esta función es la función principal del sensor, la cual se encarga de capturar la imagen de la huella en caso de que el sensor de proximidad haya detectado una yema posicionada en el cristal, después de esto la unidad DSP tratará de analizar la imagen renderizada y si esta es emparejada con alguna plantilla existente en la memoria flash, la función retornará el número del ID con el cual fue emparejada la huella.

```
int getFingerprintIDez();//leer
```

*Fig. 19 Función de obtención de huella y ID identificado. Fuente: autor*

- Función *getFingerprintEnroll*: Esta función se utiliza para almacenar una plantilla de huella dactilar de un usuario nuevo en la memoria flash del sensor.

```
uint8_t deleteFingerprint(uint8_t id);//Borrar
```

*Fig. 20 Función de guardado de plantilla nueva. Fuente: autor*

- Función *deleteFingerprint*: Esta última función se encarga de eliminar una huella almacenada en la memoria flash y dejar el ID correspondiente libre de nuevo para almacenar una plantilla nueva y asignarlo a ésta.

```
uint8_t getFingerprintEnroll();//Guardar
```

Fig. 21 Función de borrado de una plantilla existente. Fuente: autor

Una vez realizados todos los ajustes de los componentes principales y teniendo la conexión fija establecida entre ellos, las últimas etapas corresponden a la comunicación inalámbrica y potencia. Para esta comunicación se decidió utilizar dos integrados capaces de realizar la codificación y decodificación de señales de radiofrecuencia y sus respectivas antenas para transmitirlos sin conexión física alguna entre ellas, de esta manera, se realizó el diseño de los siguientes circuitos para la transmisión y recepción de los datos:

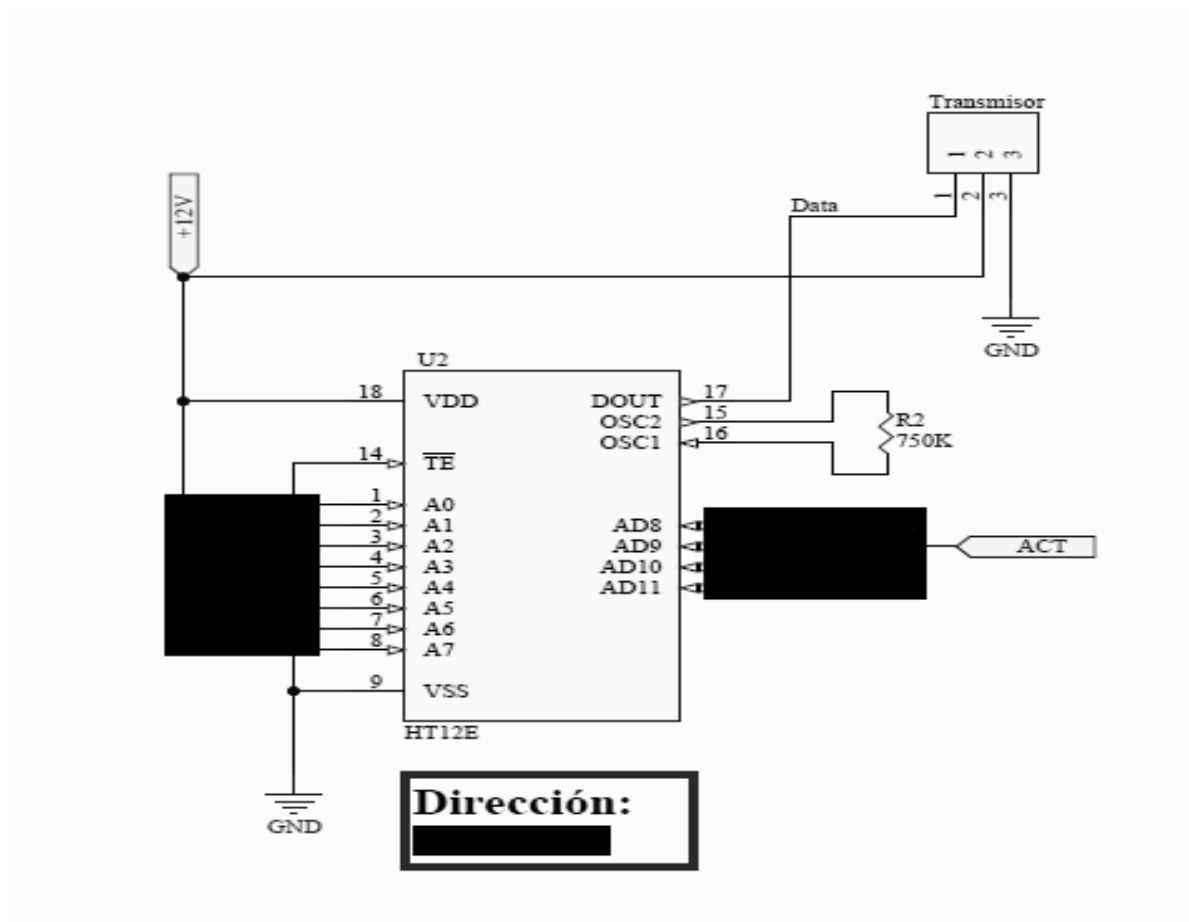


Fig. 22 Circuito emisor con antena RF. Fuente: Cristian López (integrante)

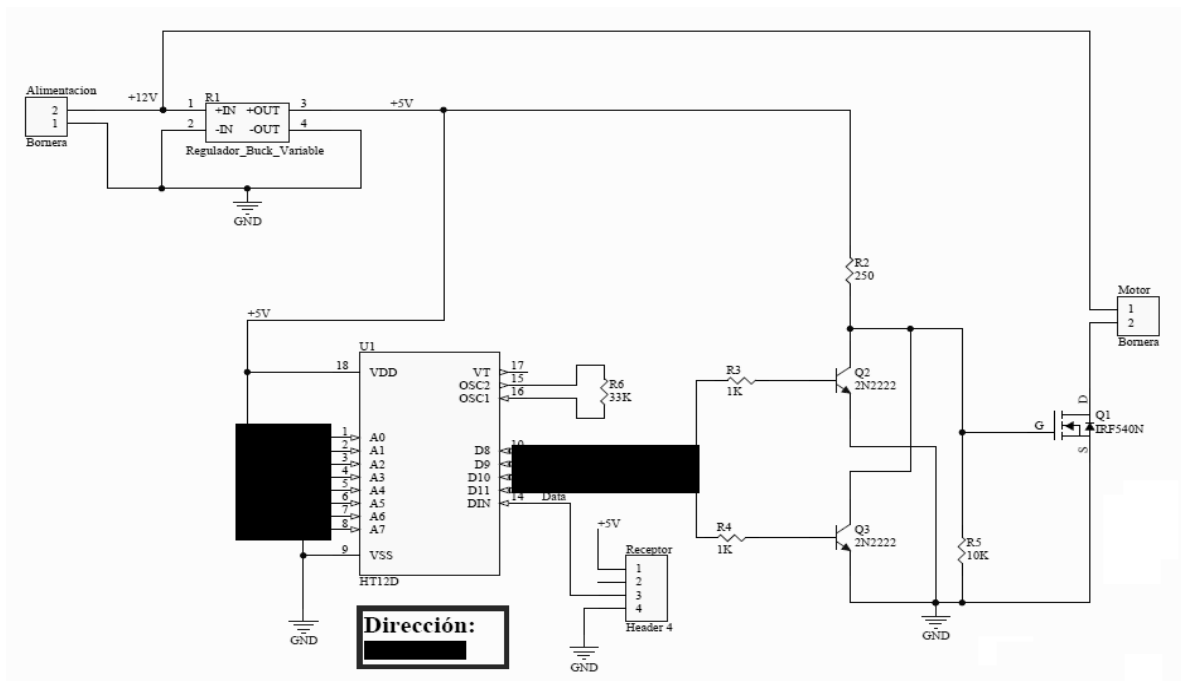
Para la etapa de potencia que va conectada con el circuito de recepción se utilizó una compuerta lógica a través de transistores tipo NPN con el fin de resolver dos situaciones que se presentaban. Primero, las señales que llegan al integrado receptor RF son de un cero lógico, por lo tanto, las entradas deben tener una resistencia pull-up constante y cuando las dos señales de entrada están en cero lógico se debe dar un 1 lógico a la salida para activar la base de un MOSFET que dará paso de corriente al motor solenoide: así la tabla de verdad que tenemos es al siguiente:

I1	I2	O
0	0	1
0	1	0
1	0	0
1	1	0

*Tabla 1 Tabla de verdad del circuito conector entre comunicación y potencia.*

*Fuente: autor*

A partir de la anterior tabla podemos deducir que lo que se necesita es una compuerta NOR, así pues, se planteó el circuito y el esquema final quedó de la siguiente manera:



*Fig. 23 Circuito receptor, compuerta lógica y etapa de potencia. Fuente: Cristian López (Integrante)*

## CAPÍTULO 4. FINALIZACIÓN DEL DISEÑO Y DISTRIBUCIÓN DE LOS SUBCIRCUITOS

Como etapa final del proyecto y teniendo definidos los requerimientos, componentes, dispositivos, integrados, diseños, esquemas, distribución y conexiones, se tienen los esquemáticos finales listos para ser exportados a un modelo virtual y concretar la etapa de diseño que es el objetivo del presente documento.

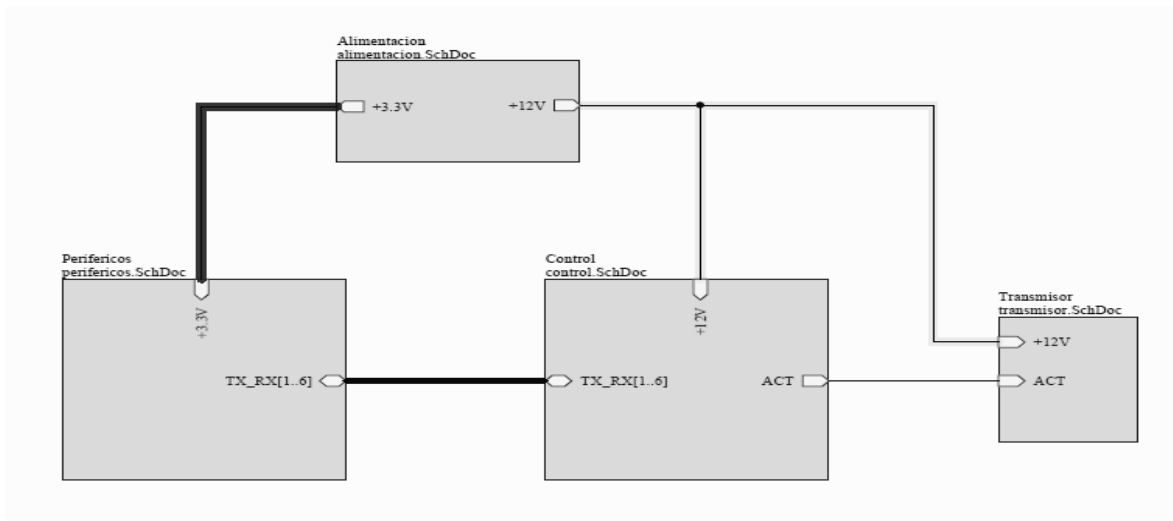


Fig. 24 Diagrama general del sistema. Fuente: Cristian López (integrante)

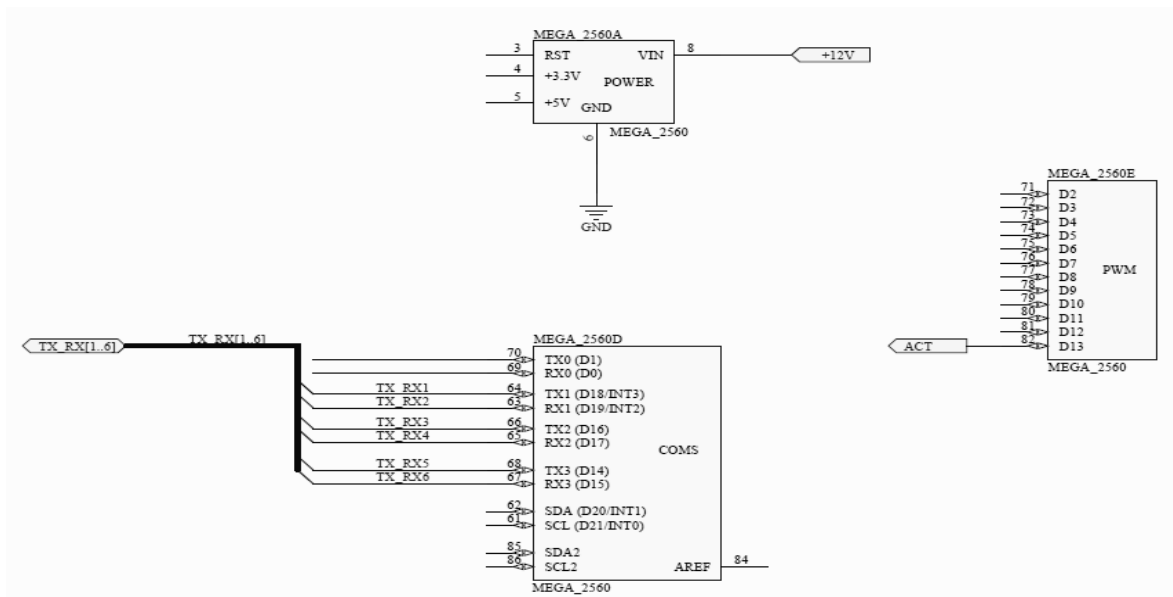
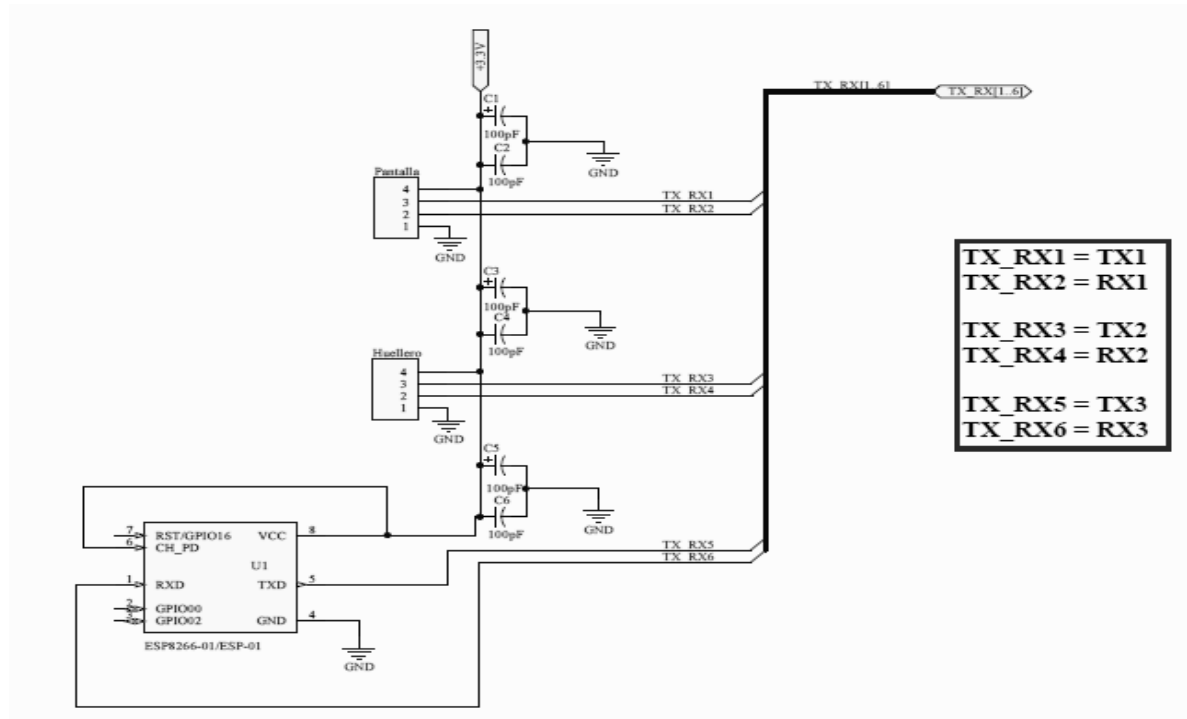


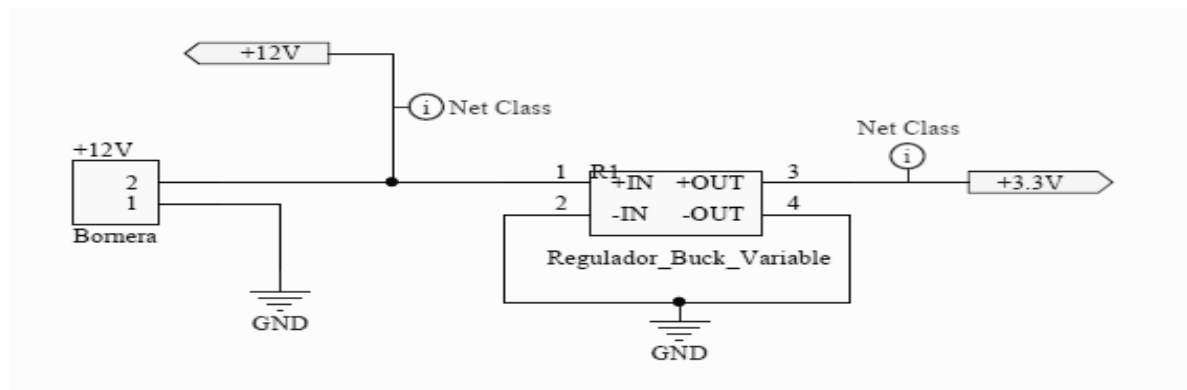
Fig. 25 Diagrama de conexiones de la placa Arduino Mega 2560. Fuente: Cristian López (integrante)

La conexión entre el Arduino y los periféricos se realiza enteramente a través de los pines de comunicación serial asíncrona (UART), y el esquema de conexiones es el siguiente:



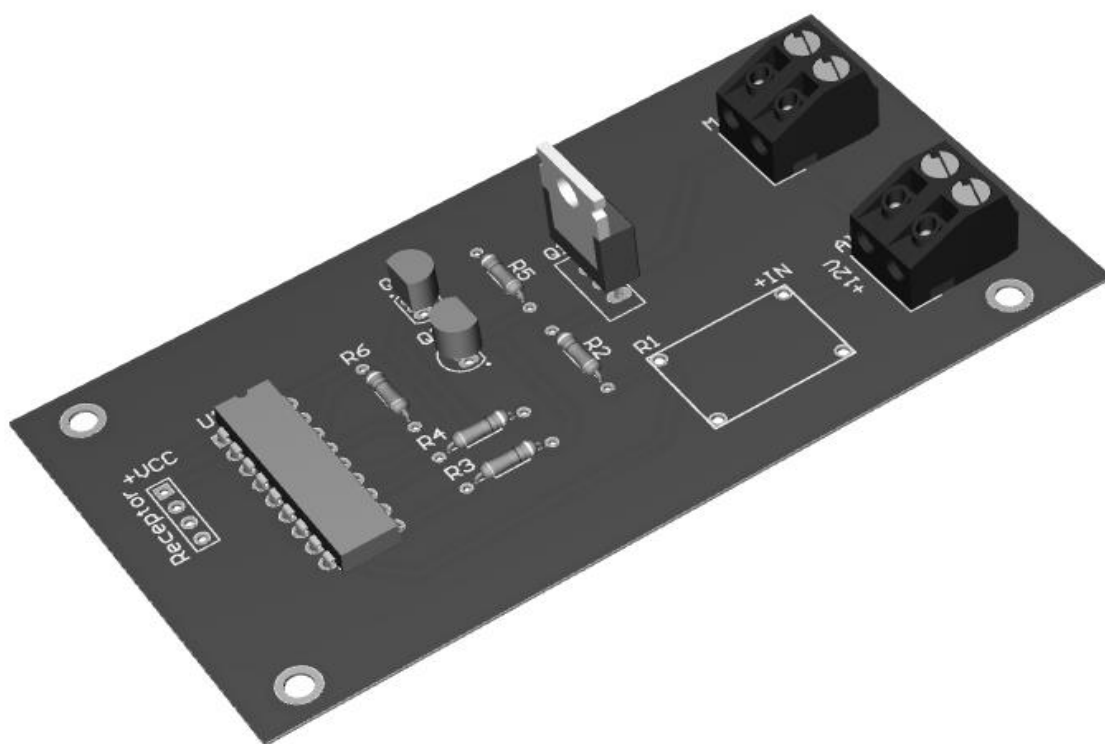
*Fig. 26 Diagrama de conexiones entre Arduino y periféricos. Fuente: Cristian López (integrante)*

Finalmente, ya que existe un rango de dispositivos a manejar, se pre-diseñó una etapa de alimentación que otorga todos los rangos de voltajes necesarios para cada componente sin limitar la corriente requerida, de la siguiente manera:



*Fig. 27 Diseño de la etapa de alimentación variable. Fuente: Cristian López (integrante)*





*Fig. 28 Ejemplo de modelo 3D de placa de recepción y potencia. Fuente: Cristian López (integrante)*

## ANÁLISIS Y CONCLUSIONES

- Durante el desarrollo del proyecto se determinó que el diseño propio de este tipo de sistemas domóticos es mejor sobre la compra de un sistema comerciable debido al nivel de intervención, customización y personalización que ofrecen dichos productos y a la relativa facilidad de construcción gracias a la cantidad de componentes en el mercado compatibles con la plataforma Arduino y que dan robustez al proyecto.
- Se concluyó que cada dispositivo usado pudo ser aprovechado al máximo gracias a las guías que ofrecen los fabricantes y los respectivos *datasheets* que contienen cada especificación y modo de operación para mantener los beneficios obtenidos al máximo.
- La adecuada selección de los componentes permitió la versatilidad del sistema y que todos los requisitos o pautas planteados inicialmente se cumplieran en su totalidad en el diseño final.
- Tener en cuenta la ubicación y localización del sistema influye de manera directa en las etapas iniciales del proyecto para escoger los componentes que mejor se adapten a dichas características.

## RECOMENDACIONES

- Es recomendable que cuando se vaya a diseñar un sistema integrado en donde distintos dispositivos o componentes se complementen entre sí, se tengan en cuenta las especificaciones de cada uno de acuerdo con sus *datasheets* y las instrucciones dadas por los fabricantes para poder asegurar la posible y correcta comunicación entre los mismos en el sistema general.
- Cuando se trabaja con dispositivos que se conectan a una red global de manera inalámbrica es muy recomendable tener en cuenta los tiempos de conexión de estos para asegurar que en una aplicación estable se estén otorgando las facilidades para realizar estas conexiones de manera satisfactoria.
- Es importante tener muy en cuenta que, al utilizar una variedad alta de componentes, cada uno puede tener unos requerimientos específicos diferentes en cuanto a los rangos de operación y de alimentación, así, tener una fuente capaz de otorgar todos los valores dentro del rango deber ser mandatorio si se quiere cubrir todos estos aspectos y alargar los tiempos de vida de cada dispositivo y por lo tanto el tiempo de vida del sistema a construir en general.

## PRESUPUESTO

### Presupuesto

No. De actividad	Descripción	Rubro	Tiempo de ejecución (hrs)	Valor unitario en COP	Valor total en COP
Actividad 1.1	Reunión inicial en laboratorio	Costo especial técnico especializado	4	\$ 6.000,00	\$24,000.00
Actividad 1.2	Arduino Mega	Costo materiales	1	\$ 35.000,00	\$35,000.00
Actividad 1.2	Módulo Wifi ESP8266-01	Costo materiales	1	\$ 12.000,00	\$12,000.00
Actividad 1.2	Sensor de huella dactilar para Arduino	Costo materiales	1	\$100.000,00	\$100,000.00
Actividad 1.2	Pantalla táctil LCD 2,4 pulgadas para Arduino	Costo materiales	1	\$ 35.000,00	\$35,000.00
Actividad 1.3	Investigación de funcionamiento	Costo especial técnico especializado	24	\$ 6.000,00	\$144,000.00
Actividad 1.4	Exposición de resultados	Costo especial técnico especializado	4	\$ 6.000,00	\$24,000.00
Actividad 2.1	Pruebas de funcionamiento	Costo especial técnico especializado	4	\$ 6.000,00	\$24,000.00
Actividad 2.2	Ajustes y cambios	Costo especial técnico especializado	24	\$ 6.000,00	\$144,000.00
Actividad 2.3	Módulos receptor y emisor RF	Costo materiales	2	\$ 2.000,00	\$4,000.00
Actividad 2.3	Antenas comunicación RF 434MHz	Costo materiales	2	\$ 10.000,00	\$20,000.00
Actividad 2.3	Resistencias	Costo materiales	10	\$ 150,00	\$1,500.00
Actividad 2.3	Transistores NPN	Costo materiales	2	\$ 200,00	\$400.00
Actividad 2.3	MOSFET de canal N	Costo materiales	1	\$ 4.000,00	\$4,000.00
Actividad 2.4	Comprobación de etapas de potencia y com.	Costo especial técnico especializado	4	\$ 6.000,00	\$24,000.00
Actividad 3.1	Diseño general del circuito	Costo especial técnico especializado	4	\$ 6.000,00	\$24,000.00
Actividad 3.2	Realización de esquemático y distribución física	Costo especial técnico especializado	8	\$ 6.000,00	\$48,000.00
				Total	\$667,900.00

## BIBLIOGRAFÍA

- [1] M. Molina-Borja and M. A. Rodríguez-Domínguez, "Evolution of biometric and life-history traits in lizards (*Gallotia*) from the Canary Islands," *J. Zool. Syst. Evol. Res.*, vol. 42, no. 1, pp. 44–53, Feb. 2004.
- [2] Berthold Laufer, "History of the Fingerprint System," *The Print*, vol. 16, no. February, pp. 1–8, 1970.
- [3] "DigitalPersona U.are.U® Fingerprint Reader Regulatory Information," p. 94063, 2007.
- [4] "Fingerprint sensor ID: 751 - \$49.95 : Adafruit Industries, Unique & fun DIY electronics and kits." [Online]. Available: <https://www.adafruit.com/product/751>. [Accessed: 14-Nov-2017].
- [5] "Control de acceso con clave." [Online]. Available: <https://www.prometec.net/control-acceso-clave/>. [Accessed: 07-Jan-2018].
- [6] "NFC RFID-RC522 Tarjeta RF RF Lector RFID módulo w / S50 tarjeta para arduino - sin Gastos de Envío - DealExtreme." [Online]. Available: <http://www.dx.com/es/p/nfc-rfid-rc522-rf-ic-card-sensor-rfid-reader-module-w-s50-card-keychain-for-arduino-403052#.WII1CFXibIU>. [Accessed: 07-Jan-2018].
- [7] SparkFun Electronics, "WiFi Module - ESP8266 - WRL-13678 - SparkFun Electronics." .
- [8] "2.2 pulgadas TFT LCD de pantalla táctil Expansión escudo para Arduino." [Online]. Available: [https://www.geekbuying.com/item/2-2-Inch-TFT-LCD-Touch-Screen-Expansion-Shield-for-Arduino-382055.html?Currency=USD&gclid=Cj0KCQiAp8fSBRCUARIsABPL6JYWuVAg5-PCNAxIduq4DgV4NS37-aaUruBy5wylpywI4QtnQkwCxDMaAIJnEALw\\_wcB#googtrans/en/es](https://www.geekbuying.com/item/2-2-Inch-TFT-LCD-Touch-Screen-Expansion-Shield-for-Arduino-382055.html?Currency=USD&gclid=Cj0KCQiAp8fSBRCUARIsABPL6JYWuVAg5-PCNAxIduq4DgV4NS37-aaUruBy5wylpywI4QtnQkwCxDMaAIJnEALw_wcB#googtrans/en/es). [Accessed: 07-Jan-2018].
- [9] D. Mellis, "Arduino Mega 2560," *Retrieved Novemb.*, p. 2560, 2011.
- [10] "Server-Client Communication using ESP8266 Serial WiFi Module (ESP-01)." [Online]. Available: <https://www.rhydolabz.com/wiki/?p=10872>. [Accessed: 10-Jan-2018].
- [11] "ARduino mega ESP8266-01." [Online]. Available: <https://forum.arduino.cc/index.php?topic=489461.0>. [Accessed: 10-Jan-2018].
- [12] "Finger print scanner (FPS) on arduino - Homautomation." [Online]. Available: <http://www.homautomation.org/2014/10/11/playing-with-finger-print-scanner-fps-on-arduino/>. [Accessed: 10-Jan-2018].

## ANEXOS

### Anexo A

#### Código de funciones del módulo Wifi:

```
#include <SoftwareSerial.h>

String Network = "CUARTOS PISO 1";

String Pass = "29571154";

String Host = "192.168.0.107";

//String Network = "CVGIRA";

//String Pass = "VisionLab";

//String Host = "192.168.1.8";

String Puerto = "80";

SoftwareSerial esp(10,11); // D10 como RX y D11 como TX

void espStart(){

    esp.begin(9600);

}

void espEnd(){

    esp.end();

}

void espOk(){

    espClear();

    esp.println("AT");

    if (esp.find("OK"))

    {

        Serial.println("Comunicación con el módulo establecida");

    }

}

void espWorkMode(){

    espClear();
```

```

esp.println("AT+CWMODE=3");

if (esp.find("OK"))
{
    Serial.println("Modos punto de acceso y estacion establecidos");
    Serial.println("");
}
}

void espWifiConnect(){
    espClear();

    esp.println("AT+CWJAP=\"" + Network + "\", \"" + Pass + "\"");
    delay(1000);
    if (esp.find("OK"))
    {
        Serial.println("");
        Serial.println("Red Wifi conectada");
    }
    else
    {
        Serial.println("Conectando a la red...");
        espWifiConnect();
    }
}

void espClear(){
    while (esp.available() > 0)
    {
        int data = esp.read();
    }
}

```

```

void espRegister(int ID){
    String var = "";
    var += (char)ID;
    String url = "GET /acceso/enviardatos.php?Op=1&ID="+var;
    int datal = url.length()+2;
    espClear();
    esp.println("AT+CIPSTART=\\"TCP\\",\\""+Host+"\\", "+Puerto);
    if (esp.find("CONNECT\r\n\r\nOK"))
    {
        Serial.println("");
        Serial.println("Conectado al servidor");
        espClear();
        esp.println("AT+CIPSEND="+ (String)datal);
        if (esp.find(">"))
        {
            Serial.println("Listo para enviar datos");
            espClear();
            esp.println(url);
            if (esp.find("RECEIVED"))
            {
                Serial.println("Dato enviado");
            }
        }
    }
    espClear();
}

void espUser(int ID, String Nombre, String Apellido){
    String url = "GET /acceso/enviardatos.php?Op=2&ID="+ (String)ID+"&N="+Nombre+"&A="+Apellido;

```



```

int datal = url.length()+2;

espClear();

esp.println("AT+CIPSTART=\"TCP\", \""+Host+"\", "+Puerto);

if (esp.find("CONNECT\r\n\r\nOK"))

{

    Serial.println("");

    Serial.println("Conectado al servidor");

    espClear();

    esp.println("AT+CIPSEND="+ (String) datal);

    if (esp.find(">"))

    {

        Serial.println("Listo para enviar datos");

        espClear();

        esp.println(url);

        if (esp.find("RECEIVED"))

        {

            Serial.println("Dato enviado");

        }

    }

}

espClear();

}

```

## **Anexo B**

### **Código del sensor de huella dactilar:**

```

#include "Hueller.h"

extern Adafruit_Fingerprint finger;

```

```

void setup() {

  Serial.begin(9600);

  finger.begin(57600);

}

int op=0 ,idm=-1;

int bandera=0;

uint8_t id=0;

void loop() {

  getFingerprintIDez();

  if(Serial.available()){

    op=(Serial.readStringUntil(' ')).toInt();

    bandera=op;

  }

  switch (op){

  case 2:

    {

      Serial.println("Huella Maestra");

      while (idm== -1){

        idm=getFingerprintIDez();

      }

      Serial.println(idm);

      if(idm==2){

        Serial.println("ID guardar");

        while(op==bandera){

          if(Serial.available()){

            op=(Serial.readStringUntil(' ')).toInt();

            bandera=op;

            break;

          }

        }

      }

    }

  }

}

```

```

    }
}

id=op;

Serial.println(id);

getFingerprintEnroll();

idm=-1;
}

else

{

idm=-1;

op=0;

bandera=0;

Serial.println("Invalida");

break;

}

break;

}

break;

case 3:

{

Serial.println("Huella Maestra");

while (idm==1){

idm=getFingerprintIDez();

}

Serial.println(idm);

if(idm==2){

Serial.println("ID borrar");

while(op==bandera){

```

```

        if(Serial.available()){
            op=(Serial.readStringUntil(' ')).toInt();

            bandera=op;

            break;

        }

    }

    id=op;

    Serial.print(id); Serial.println(" Borrada");

    deleteFingerprint(id);

}

else

{

    idm=-1;

    op=0;

    bandera=0;

    Serial.println("Invalida");

    break;

}

break;

}

break;

default:

{

    getFingerprintIDez();

}

break;

}

}

```

## Anexo c

Código .php del servidor que redirige la información a la base de datos:

```
<?php

$conexion = @mysql_connect("localhost", "*****", "*****");

mysql_select_db("acceso", $conexion);

$ID = $_GET ['ID'];

if ($_GET ['Op'] == 1) // Operación de registro de visita
{
    list($N, $A) = mysql_fetch_array(mysql_query("SELECT `Nombre`, `Apellido` FROM `acceso`.`usuarios`
    WHERE `User ID` = $ID"), MYSQL_NUM);

    mysql_query("INSERT INTO `acceso`.`registros` (`User ID`, `Nombre`, `Apellido`) VALUES
    ('$ID', '$N', '$A')");

    //echo ("$N $A");

    echo ("RECEIVED");
}

else if ($_GET ['Op'] == 2) // Operación de ingreso de un nuevo usuario
{
    $N = $_GET ['N'];

    $A = $_GET ['A'];

    mysql_query("INSERT INTO `acceso`.`usuarios` (`User ID`, `Nombre`, `Apellido`) VALUES
    ('$ID', '$N', '$A')");
}

mysql_close();

?>
```